**SVM**

May 2007

**DOE-PI**

**Dianne P. O'Leary**

# Speeding the Training of Support Vector Machines and Solution of Quadratic Programs

**Dianne P. O'Leary**
Computer Science Dept. and
Institute for Advanced Computer Studies
University of Maryland

**Jin Hyuk Jung**
Computer Science Dept.

**André Tits**
Dept. of Electrical and Computer Engineering and
Institute for Systems Research

Work supported by the Department of Energy.

# The Plan

- Very brief overview of our work
- Introduction to SVMs
- Our algorithm
- Examples

## Our DOE-supported research program

Goal: Develop efficient algorithms for optimization problems having a large number of inequality constraints.

Example applications:

- Semi-infinite programming: problems with pde constraints.
- Training support vector machines.

## Progress:

- Efficient implementation of an interior point method (IPM) for solving linear programs on a GPU (= graphical processing unit).
  (Jung, O'Leary) (poster)
- Adaptive constraint reduction algorithms for linear (poster) and quadratic programming problems.
  (Stacey Nicholls, Luke Winternitz, Jung, O'Leary, Tits)
- Simple conditions on the "constraint matrices" and cone for a pair of dual conic convex programs, under which the duality gap is zero for *every* choice of linear objective function and "right-hand-side".
  (Simon Schurr, O'Leary, Tits)
- Convex duality and entropy-based closure in gas dynamics.
  (Cory Hauck, Tits, David Levermore) (poster)
- A polynomial-time interior-point method for conic optimization, with inexact barrier evaluations. (Schurr, O'Leary, Tits) (poster)
- SVM training (Jung, O'Leary, Tits) (this talk)

## The problem

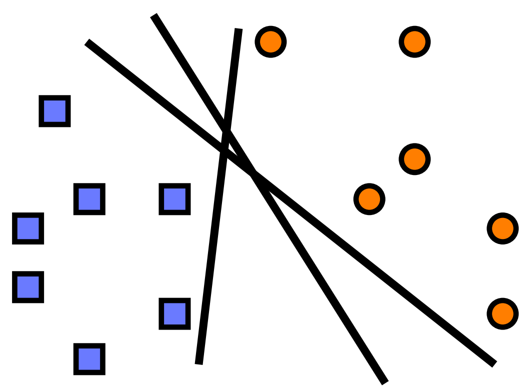
Given: A set of sample data points $\mathbf{a}_i$, in sample space $\mathcal{S}$, with labels $d_i = \pm 1$, $i = 1, \ldots, m$.

Find: A hyperplane $\{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle - \gamma = 0\}$, such that

$$\operatorname{sign}(\langle \mathbf{w}, \mathbf{a}_i \rangle - \gamma) = d_i,$$

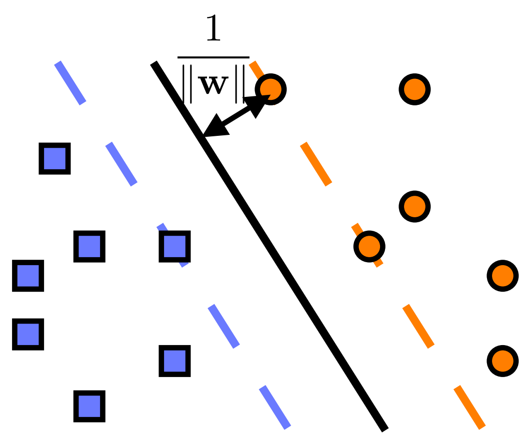
or, ideally,

$$d_i(\langle \mathbf{w}, \mathbf{a}_i \rangle - \gamma) \geq 1.$$



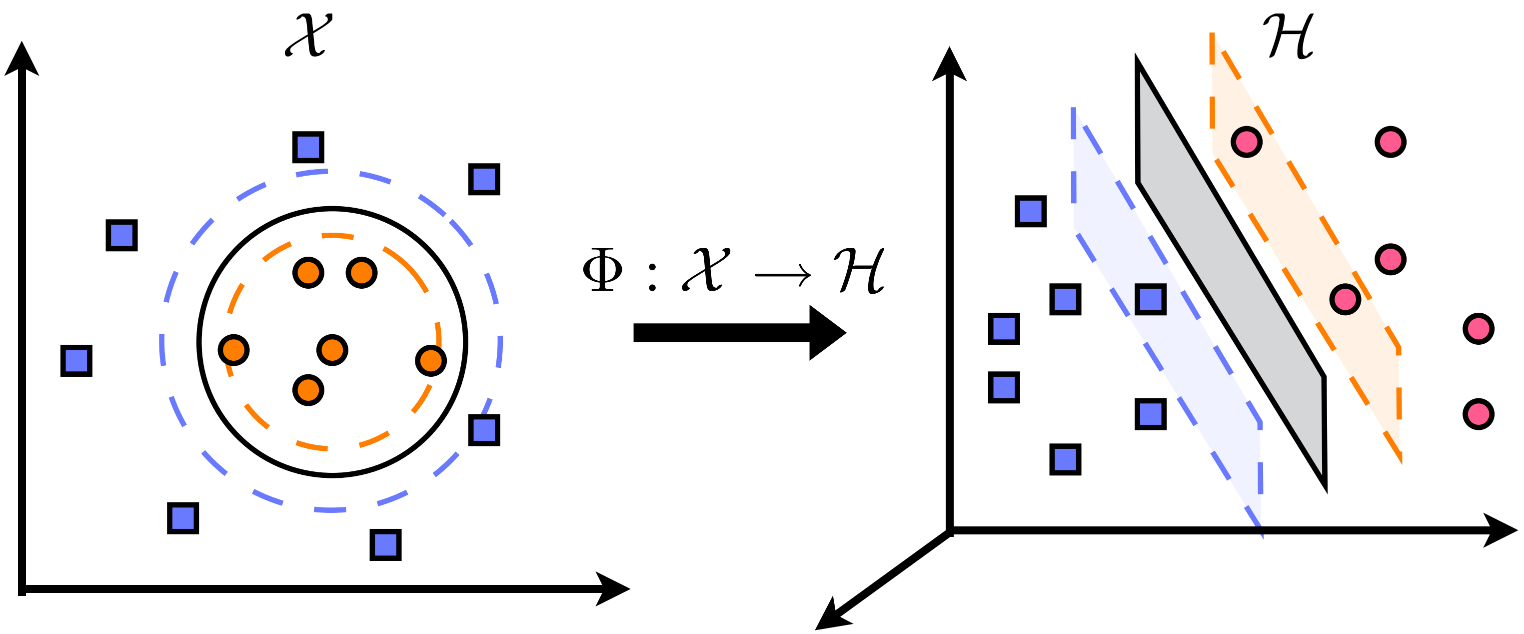
## Which hyperplane is best?

We want to maximize the separation margin $1/\|\mathbf{w}\|$.



## Generalization 1

We might map a more general separator to a hyperplane through some transformation $\Phi$:




For simplicity, we will assume that this mapping has already been done.

## Generalization 2

If there is no separating hyperplane, we might want to balance maximizing the separation margin with a penalty for misclassifying data by putting it on the wrong side of the hyperplane. This is the soft-margin SVM.

- We introduce slack variables $\mathbf{y} \geq \mathbf{0}$ and relax the constraints $d_i(\langle \mathbf{w}, \mathbf{a}_i \rangle - \gamma) \geq 1$ to

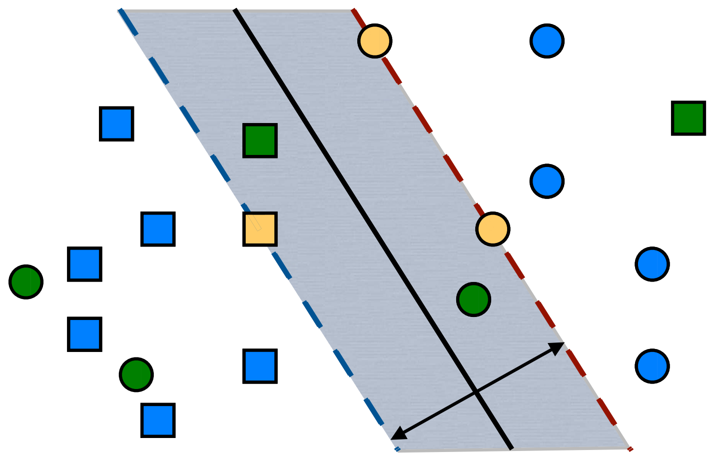
$$d_i(\langle \mathbf{w}, \mathbf{a}_i \rangle - \gamma) \geq 1 - y_i.$$

- Instead of minimizing $\|\mathbf{w}\|$, we solve

$$\min_{\mathbf{w},\gamma,\mathbf{y}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \tau \mathbf{e}^T \mathbf{y}$$

for some $\tau > 0$, subject to the relaxed constraints.

## Jargon



Classifier $\langle \mathbf{w}, \mathbf{x} \rangle - \gamma = 0$: black line
Boundary hyperplanes: dashed lines
2 $\times$ separation margin: length of arrow
Support vectors: On-Boundary (yellow) and Out-of-Bound (green)
Non-SV: blue

> Key point: The classifier is the same, regardless of the presence or absence of the blue points.

## More jargon

- The process of determining $\mathbf{w}$ and $\gamma$ is called training the machine.
- After training, given a new data point $\mathbf{x}$, we simply calculate $\operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle - \gamma)$ to classify it as in either the positive or negative group.
- This process is thought of as a machine – called the support vector machine (SVM).
- We will see that training the machine involves solving a convex quadratic programming problem whose number of variables is the dimension $n$ of the sample space and whose number of constraints is the number $m$ of sample points – typically very large.

# Primal and dual

Primal problem:

$$
\begin{aligned}
\min_{\mathbf{w},\gamma,\mathbf{y}} \ & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + \tau \mathbf{e}^T\mathbf{y} \\
s.t. \quad & \mathbf{D}(\mathbf{A}\mathbf{w} - \mathbf{e}\gamma) + \mathbf{y} \geq \mathbf{e}, \\
& \mathbf{y} \geq \mathbf{0},
\end{aligned}
$$

Dual problem:

$$
\begin{aligned}
\max_{\mathbf{v}} \ & -\tfrac{1}{2}\mathbf{v}^T\mathbf{H}\mathbf{v} + \mathbf{e}^T\mathbf{v} \\
s.t. \quad & \mathbf{e}^T\mathbf{D}\mathbf{v} = 0, \\
& \mathbf{0} \leq \mathbf{v} \leq \tau\mathbf{e},
\end{aligned}
$$

where $\mathbf{H} = \mathbf{D}\mathbf{A}\mathbf{A}^T\mathbf{D} \in \mathbb{R}^{m\times m}$ is a symmetric and positive semidefinite matrix with

$$h_{ij} = d_i d_j \langle \mathbf{a}_i, \mathbf{a}_j \rangle.$$

## Support vectors

Support vectors (SVs) are the patterns that contribute to defining the classifier.

They are associated with nonzero $v_i$.

| | $v_i$ | $s_i$ | $y_i$ |
|---|---|---|---|
| Support vector | $(0, \tau]$ | 0 | $[0, \infty)$ |
| On-Boundary SV | $(0, \tau)$ | 0 | 0 |
| Out-of-Bound SV | $\tau$ | 0 | $(0, \infty)$ |
| Nonsupport vector | 0 | $(0, \infty)$ | 0 |

- $v_i$: dual variable (Lagrange multiplier for relaxed constraints).
- $s_i$: slack variable for nonnegativity of $v_i$; i.e., $s_i v_i = 0$.
- $y_i$: slack variable in relaxed constraints.

## Solving the SVM problem

Apply standard optimization machinery:

- Write down the optimality conditions for the primal/dual formulation using the Lagrange multipliers. This is a system of nonlinear equations.
- Apply a (Mehotra-style predictor-corrector) interior point method (IPM) to solve the nonlinear equations by tracing out a path from a given starting point to the solution.

- At each step of the IPM, the next point on the path is computed using a variant of Newton's method by solving the linear system of equations

$$\mathbf{M}\ \Delta\mathbf{w} = \text{some vector}$$

(sometimes called the normal equations), where

$$\mathbf{M} = \mathbf{I} + \mathbf{A}^T\mathbf{D}\mathbf{\Omega}^{-1}\mathbf{D}\mathbf{A} - \frac{\bar{\mathbf{d}}\bar{\mathbf{d}}^T}{\mathbf{d}^T\mathbf{\Omega}^{-1}\mathbf{d}}.$$

Here, $\mathbf{D}$ and $\mathbf{\Omega}$ are diagonal, $\bar{\mathbf{d}} = \mathbf{A}^T\mathbf{D}\mathbf{\Omega}^{-1}\mathbf{d}$, and

$$\omega_i^{-1} = \frac{v_i(\tau - v_i)}{s_i v_i + y_i(\tau - v_i)}.$$

Examining $\mathbf{M} = \mathbf{I} + \mathbf{A}^T \mathbf{D} \boldsymbol{\Omega}^{-1} \mathbf{D} \mathbf{A} - \dfrac{\bar{\mathbf{d}} \bar{\mathbf{d}}^T}{\mathbf{d}^T \boldsymbol{\Omega}^{-1} \mathbf{d}}$

Our approach is to modify Newton's method by using an approximation to the last two terms.

Note that the middle term is

$$\mathbf{A}^T\mathbf{D}\mathbf{\Omega}^{-1}\mathbf{D}\mathbf{A} = \sum_{i=1}^{m} \frac{1}{\omega_i}\mathbf{a}_i\mathbf{a}_i^T.$$

We only include certain terms corresponding to the large values of $\omega_i^{-1}$.
We could choose:

- patterns $\mathbf{a}_i$ with smallest distance to the class boundary hyperplanes.
- patterns $\mathbf{a}_i$ with smallest "one-sided" distance to these hyperplanes.
- patterns with largest $\omega_i^{-1}$.

We could

- ignore the value of $d_i$.
- balance the number of positive and negative patterns included.

## Some related work

- Use of approximations to $\mathbf{M}$ in LP-IPMs dates back to Karmarkar (1984), and adaptive inclusion of terms was studied, for example, by Wang and O'Leary (2000).
- Osuna, Freund, and Girosi (1997) proposed solving a sequence of CQPs, building up patterns as new candidates for support vectors are identified.
- Joachims (1998) and Platt(1999) used variants related to Osuna et al.
- Ferris and Munson (2002) focused on efficient solution of normal equations.
- Gertz and Griffin (2005) used preconditioned cg, with a preconditioner based on neglecting terms in $\mathbf{M}$.

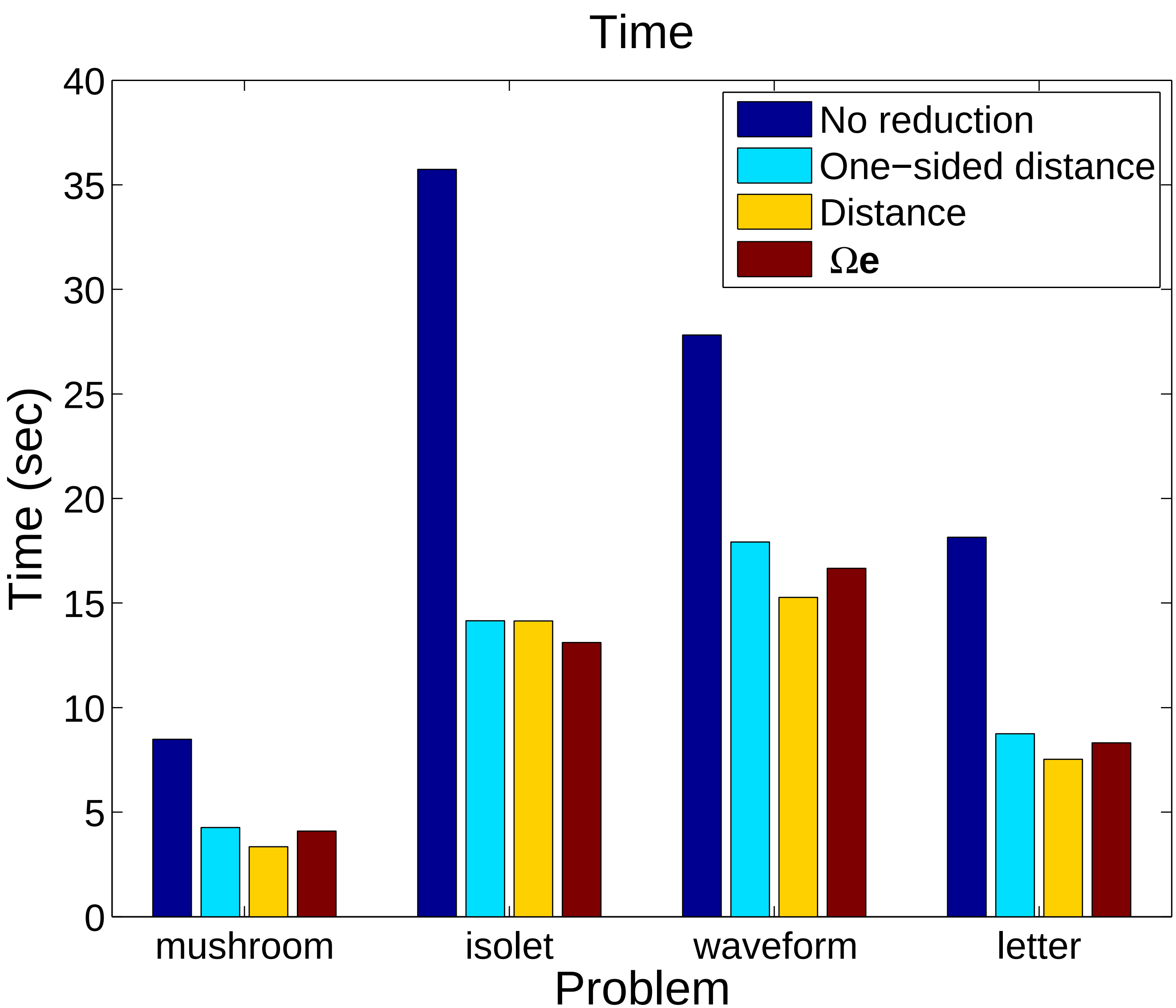
## Test problems

Provided by Josh Griffin (SANDIA)

| Problem | $n$ | Patterns $(+,-)$ | SV $(+,-)$ | In-bound SVs $(+,-)$ |
|---|---|---|---|---|
| mushroom | 276 | (4208,3916) | (1146,1139) | (31,21) |
| isolet | 617 | (300,7497) | (74,112) | (74,112) |
| waveform | 861 | (1692,3308) | (633,638) | (110,118) |
| letter-recog | 153 | (789,19211) | (266,277) | (10,30) |

The number of iterations was almost constant, regardless of algorithm variant, so we measure time for solution (MATLAB).

We used the balanced selection scheme.


Time
No reduction
One−sided distance
Distance
Ωe
Time (sec)
0
5
10
15
20
25
30
35
40
mushroom
isolet
waveform
letter
Problem

## Comparison with other software

| Problem | Type | LibSVM | SVMLight | Matlab | Ours |
|---|---|---|---|---|---|
| mushroom | Polynomial | 5.8 | 52.2 | 1280.7 | |
| mushroom | Mapping(Linear) | 30.7 | 60.2 | 710.1 | 4.2 |
| isolet | Linear | 6.5 | 30.8 | 323.9 | 20.1 |
| waveform | Polynomial | 2.9 | 23.5 | 8404.1 | |
| waveform | Mapping(Linear) | 33.0 | 85.8 | 1361.8 | 16.2 |
| letter | Polynomial | 2.8 | 55.8 | 2831.2 | |
| letter | Mapping(Linear) | 11.6 | 45.9 | 287.4 | 13.5 |

- LibSVM, by Chih-Chung Chang and Chih-Jen Lin, uses a variant of SMO (by Platt), implemented in C
- SVMLight, by Joachims, implemented in C
- Matlab's provided program is a variant of SMO.
- Our program is implemented in Matlab, so we would expect a speed-up if converted to C.

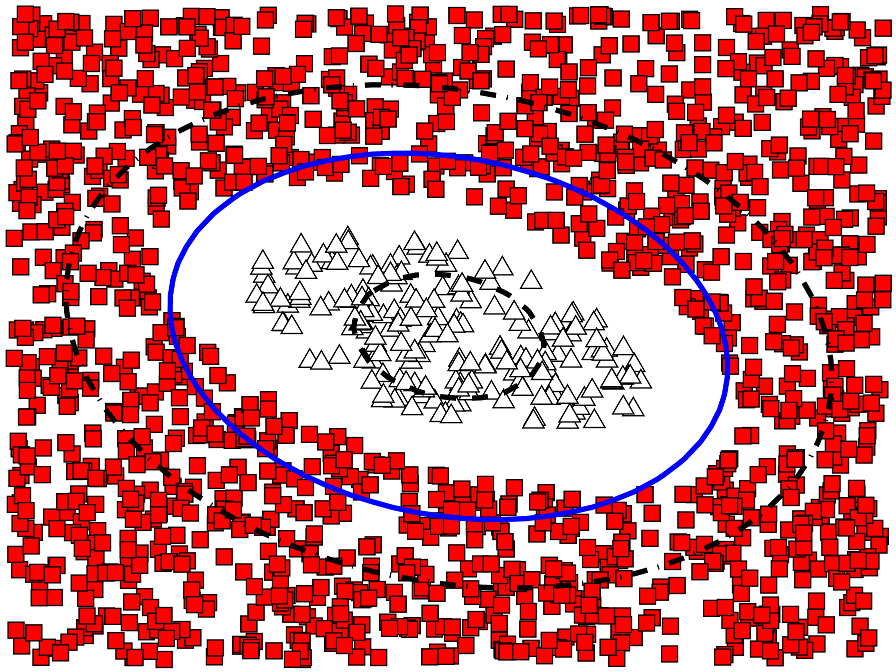
## How our algorithm works

To visualize the iteration, we constructed a toy problem with

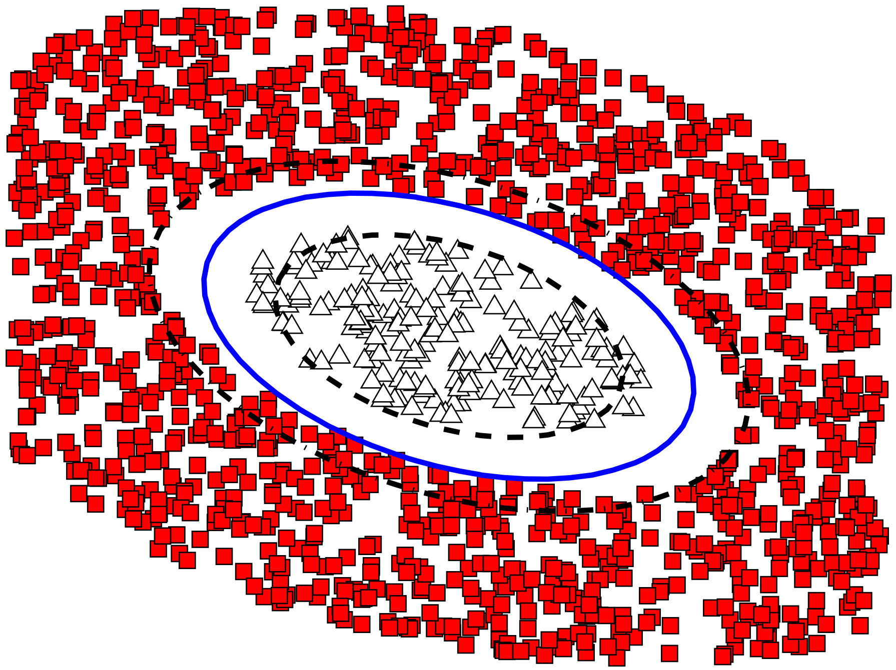
- $n = 2$,
- a mapping $\Phi$ corresponding to an ellipsoidal separator.

We now show snapshots of the patterns that contribute to $\mathbf{M}$ as the IPM iteration proceeds.

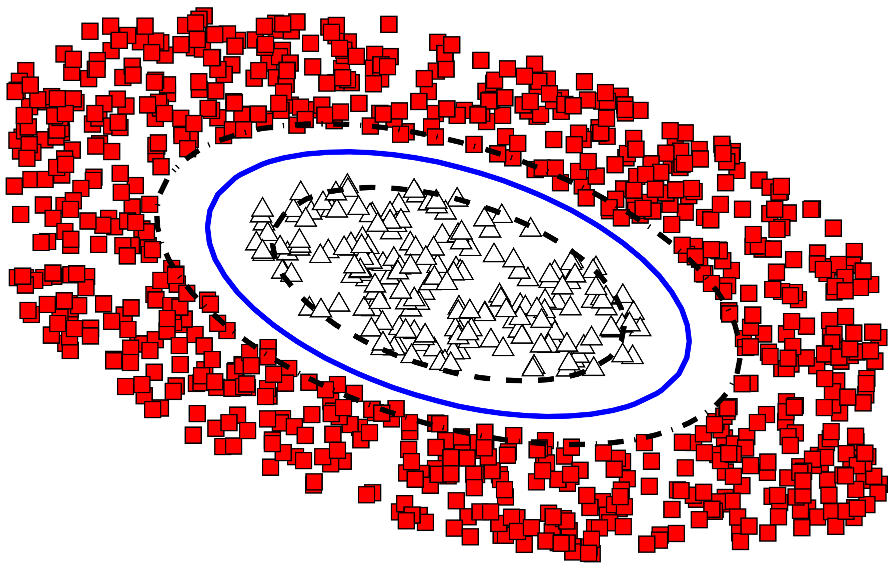
Iteration:  2, # of obs: 1727



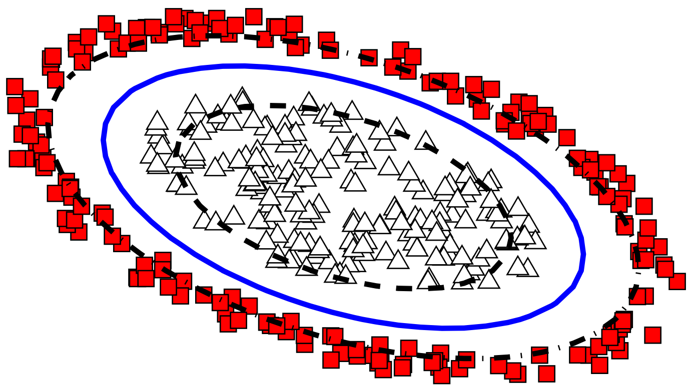
Iteration:  5, # of obs: 1440



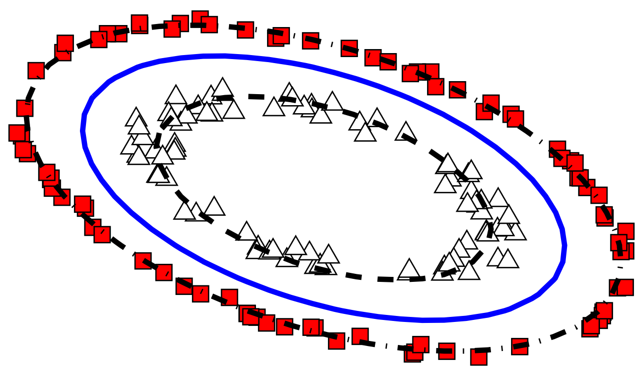
Iteration:  8, # of obs: 1026



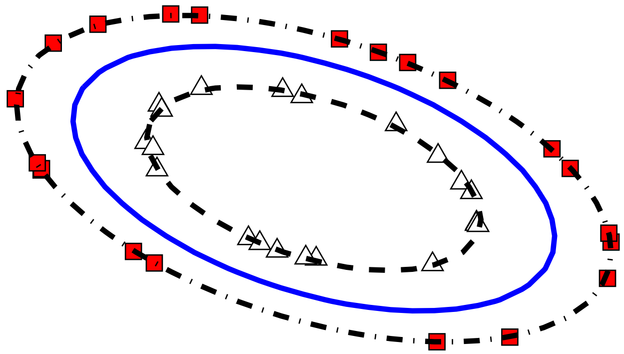
Iteration: 11, # of obs:  376



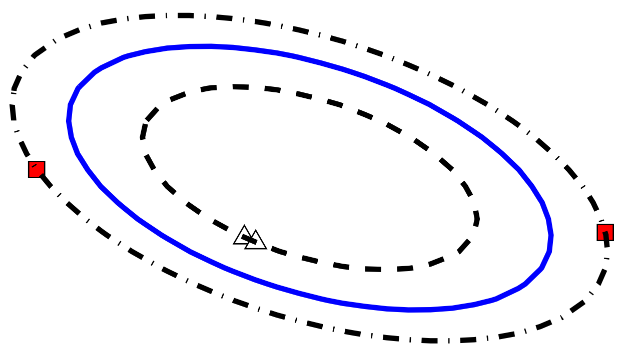
# Iteration: 14, # of obs:  170



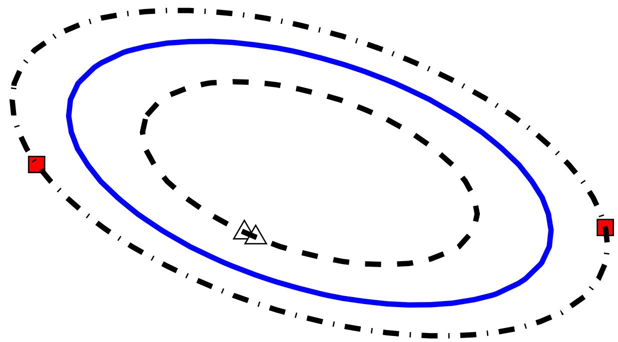
Iteration: 17, # of obs: 42



# Iteration: 20, # of obs:    4



Iteration: 23, # of obs: 4



## Conclusions

- We have succeeded in significantly improving the training of SVMs that have large numbers of training points.
- Similar techniques apply to general CQP problems with a large number of constraints.
- Savings is primarily in later iterations. Future work will focus on using clustering of patterns (e.g., Boley and Cao (2004)) to reduce work in early iterations.
- We are seeking additional classification problems of interest to DOE.